SYSTEM FOR DATA TRANSFER THROUGH AN I/O DEVICE USING A MEMORY ACCESS CONTROLLER WHICH RECEIVES AND STORES INDICATION OF A DATA STATUS SIGNAL

RELATED APPLICATIONS

This application is a continuation-in-part of U.S. Application No. 08/436,969, filed May 8, 1995, and entitled "Communication Interface with Improved Packet Transfer, now abandoned."

FIELD

The present invention relates to a method of and apparatus for using the status bits of a channel status register of a direct memory access (DMA) channel to control channel operation. It more particularly relates to the use of status bits to indicate when an external device has completed a data transmission and to indicate an end of a received data packet.

BACKGROUND

DMA channels transfer data to and from main system memory with minimal burden on the system CPU. Each channel controls itself and any attached I/O devices by executing instructions called channel commands. Channel commands are organized into very simple programs called channel programs, also referred to as a "buffer descriptor list" in a descriptor based DMA system such as the DBDMA system created by Apple Computer, Inc., the assignee of the present application.

A DMA engine or controller may control operation of one or more channels. The DMA controller may contain one register file which stores the current channel program pointers for each channel, and another register file which contains the current context (address, count, and flags) for each of the channels. An example of such an implementation is the Apple Grand Central I/O Controller which services eleven DMA channels.

The function of the DMA controller is to arbitrate data transfer requests from various I/O devices in the system. These devices may operate only in slave mode DMA, that is, they may make requests which indicate that they have data in receive registers which is available to be received by the system, or that they have space available in transmission registers for data to be sent by the system. Also, the I/O device may indicate that either a transmit or receive operation is complete by the generation of status information such as an interrupt which would normally require the attention of host processor.

It would be beneficial to have a DMA controller which could handle data and status information without calling on the host processor. In particular, it would be useful to have a DMA controller which is able to process multiple complete transmit or receive operations without calling on the host processor.

SUMMARY

It is an object of the present invention to overcome the deficiencies of the prior art by using status bits to facilitate the transfer and efficient storage of packets of data.

This object is achieved through the provision of a method of transferring a data unit between a computer system memory and an external system through an I/O device using a memory access controller where the memory access controller includes a register for storing status information which the memory access controller uses to control its own operation. A "data unit" is a discrete grouping of data such as would normally be transferred all at a time, such as a package of data in an Ethernet application. The memory access controller together with the I/O device transfer the data unit between the computer system memory and the external system. This may be a transmit operation in which the data unit is sent from the memory to the external system, or a receive operation in which the data unit is received from the external system. Upon completion of the operation, the I/O device sends a data status signal to the memory access controller. The memory access controller then stores an indication of the data status signal.

More specifically, in the case of an example of a data transmit operation, the memory access controller retrieves the data unit from the computer system memory and transmits the data unit to the I/O device. The I/O device then transmits the data unit to the external system, and, upon successful completion of this task, sends the data status signal which indicates a successful transmission of data unit to the external system by the I/O device. The memory access controller stores an indication of the data status signal by setting a bit in its channel status register. The memory access controller waits until the bit reflects a value of the data status signal indicating successful transmission of a data unit to the external system before taking additional action such as requesting status information or sending another data unit.

In the case of an example of a receive operation, the I/O device transmits the data unit as received from the external system to the memory access controller. It then sends a data status signal indicating forwarding of an end of the data unit from the external system. In response, the memory access controller sets a bit in its channel status register and requests data unit information form the I/O device. The memory access controller then appends the data unit information to the end of the received data unit. This permits flexible storage.

The external system may be a network and the I/O device may be an Ethernet controller, in which case the data unit corresponds to a data packet. The data status signal is preferably an interrupt signal.

In one particular embodiment, the memory access controller is a DMA channel controller, and there are two DMA channels, a receive channel and a transmit channel, with the interrupt from the receive channel being masked.

In the case of a receive operation, the data status signal may be an end-of-packet signal. The data unit is stored in at least one of a plurality of buffers allocated in the system memory with each of the plurality of buffers being smaller than a maximum data unit size. The data unit information preferably comprises a plurality of status bytes.

In another aspect, a computer system embodying the invention preferably includes a computer system memory and an I/O device connected to an external system to transfer data units between the computer system memory and an external system. The I/O device including a data status signal generator generates a data status signal upon completion of transfer of a data unit. A memory access controller connected to the computer system memory and the I/O device includes a register for storing status information which the memory access controller uses to control its own operation. The memory access controller receives the data status signal and stores an indication of a value of the data status signal in the register. In a transmit operation, the data status signal indicates a successful transmission of data unit

to the external system by the I/O device. During a data receiving operation the I/O device transmits the data unit received from the external system to the memory access controller. The data memory access controller uses the data status signal to append data unit information to the end of the data unit.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, with are incorporated in and constitute a part of the specification, schematically illustrate a preferred embodiment of the present invention and, together with the general description given above and the detailed description of the preferred embodiment given below, serve to explain the principles of the present invention, in which:

FIG. **1** is a functional block diagram of a computer system according to one embodiment of the present invention;

FIG. **2** illustrates organization of information in a channel control register according to one embodiment of the present invention;

FIG. **3** illustrates organization of information in a channel status register according to one embodiment of the present invention;

FIG. **4** is a functional block diagram of an I/O controller used with an Ethernet controller in accordance with one embodiment of the present invention;

FIG. **5** is a flowchart illustrating the operation of an Ethernet transmit channel according to a presently preferred embodiment of the invention;

FIG. **6** is a chart illustrating operation of an an Ethernet transmit channel according to a presently preferred embodiment of the invention;

FIG. **7** illustrates designations of Ethernet transmit channel status bits according to one embodiment of the present invention;

FIG. **8** is a flowchart illustrating the operation of the Ethernet receive channel; and

FIG. **9** illustrates designations of Ethernet receive channel status bits according to one embodiment of the present invention.

DETAILED DESCRIPTION

FIG. **1** illustrates an integrated I/O controller **10** in a system according to one embodiment of the present invention. A plurality of I/O devices or external device controllers **20** are connected through a plurality of interfaces **30** to a bus **40** within the I/O controller **10**. Each I/O device is in turn connected to an external device system.**50**. The I/O controller **10** also contains a DMA engine **60** and a bus interface **70** which are used to control the flow of data from the I/O devices **20** out onto an external system bus **80** where the data can be stored in a memory **90** or used by other devices. A DMA controller is more completely described in U.S. patent application Nos. 08/340,248 and 08/340,249, both of which are incorporated herein by reference.

The I/O controller **10** includes a channel control register **100** and a channel status register **110** which are mapped onto system memory **90**. The channel control register **100** contains bits which control the operational state of the DMA channel. One organization scheme for the information stored in the channel control register **100** is illustrated in FIG. **2**. As can be seen, the channel control register **100** is divided into two fields: a mask field **120** and a data field **130**. The mask field **120** selects which of the bits in the data field **130** are modified by a write operation.

The channel status register **110** is illustrated in FIG. **3**. It provides access to the DMA channel's internal status bits. In other words, the eight general purpose status bits are written through the channel control register and read through the channel status register. The bits s7–s0 can be used for general purpose status operations. DMA tests these bits after the completion of each channel command to determine if various actions should be taken by the DMA channel. These status bits thus can be used for general purpose status and control. Their meaning may be channel specific, and they can be controlled either by hardware or software. They may effect conditional interrupt, branch, and wait conditions.

In the present invention, one or more of the I/O devices **20** is capable of sending status information back to the I/O controller **10**. This will now be discussed for the example where the I/O device **20** is an Ethernet controller. It will be understood by one of ordinary skill in the art, however, that the invention may be embodied in connection with other I/O devices.

Turning to FIG. **4**, a system is disclosed wherein one of the I/O interfaces is embodied as an Ethernet I/O module **130**, and wherein one of the I/O devices is embodied as an Ethernet controller **140** which connects to an external Ethernet network **150**. The Ethernet controller **140** performs all of the low-level protocol handling and provides first-in first-out (FIFO) memory for data buffers in the form of a receive FIFO **160** and a transmit FIFO **170**. The Ethernet controller **140** has a slave mode DMA interface **180**, with separate request lines for transmit and receive.

When there is space to put additional data in the transmit FIFO **170**, the Ethernet controller **140** asserts a DMA request. Similarly, the Ethernet controller **140** asserts a receive DMA request when data is available in the receive FIFO **160**.

The Ethernet controller **140** is a register based peripheral. It is configured using simple memory mapped I/O read or write commands. From a software standpoint, the Ethernet controller **140** operates as a data conduit to and from the Ethernet network **150** through a register based function program, DMA data movement, and interrupt driven event handling.

As mentioned, the Ethernet controller **140** is connected to the I/O controller **10** through an Ethernet I/O module **130**. (Additional elements within the I/O controller **10** are the same as those shown in FIG. **1** and are omitted from FIG. **4** for clarity.) The Ethernet controller **140** is also connected to an external media network **150** through a media interface **190** in a media access controller **200**. The media access controller **200** moves data to the receive FIFO **160** and the transmit FIFO **170**. The Ethernet controller **140** communicates with the Ethernet I/O module **130** over DMA request lines **210**, address lines **220**, a chip select line **230**, and an interrupt line **240**.

The Ethernet I/O module **130** in the I/O controller **10** provides the necessary interfaces to the Ethernet controller **140**. When the host accesses the Ethernet controller **140** registers using memory mapped reads or writes, the Ethernet I/O module **130** performs the read or write cycle directly with the Ethernet controller **140**.

The I/O controller **10** has two DMA channels for Ethernet: a receive channel and a transmit channel. During Ethernet receive and transmit DMA operations, the Ethernet I/O module **130** moves data in and out of the Ethernet controller **140**'s FIFOs **160** and **170**, respectively, and retrieves the status information from the Ethernet controller **140** after DMA is completed. The DMA request from the